Foreign Subsystem Integration

Engineering Guidelines

Document no.	**e1144**
Edition	01.94
Supersedes	-
Manual	**Applications**
Section	6

TABLE OF CONTENTS

1 INTRODUCTION 5

1.1 Related Documentation 6

2 PHYSICAL LEVEL 7

2.1 Serial line characteristics 7

2.2 Architectures 7

3 DATA LINK LEVEL 9

4 APPLICATION LEVEL 10

4.1 Subsystem information model 10

4.2 Point database 11

4.3 Point states 11

4.4 Subsystem model design 13

5 LMS MODULAR SUBSYSTEM HANDLING 14

5.1 LMS modular presentation of database information 14

5.2 LMS modular process control 15

5.2.1 Control commands 15

5.2.2 Event acknowledgement and reset 15

6 CDSF PROTOCOL 19

6.1 "Change of state" frame 19

6.2 "Control command" frame 20

7 ANNEX A - EXAMPLES OF SUBSYSTEM MODELLING 21

7.1 A1. Fire detection unit 21

7.2 A2. CCTV unit 21

8 ANNEX B - EXAMPLES OF CONTROL COMMAND LISTS 22

8.1 B1. Fire detection unit 22

8.2 B2. CCTV unit 22

9 ANNEX C - SUBSYSTEM MODEL DESIGN CHECKLIST 23

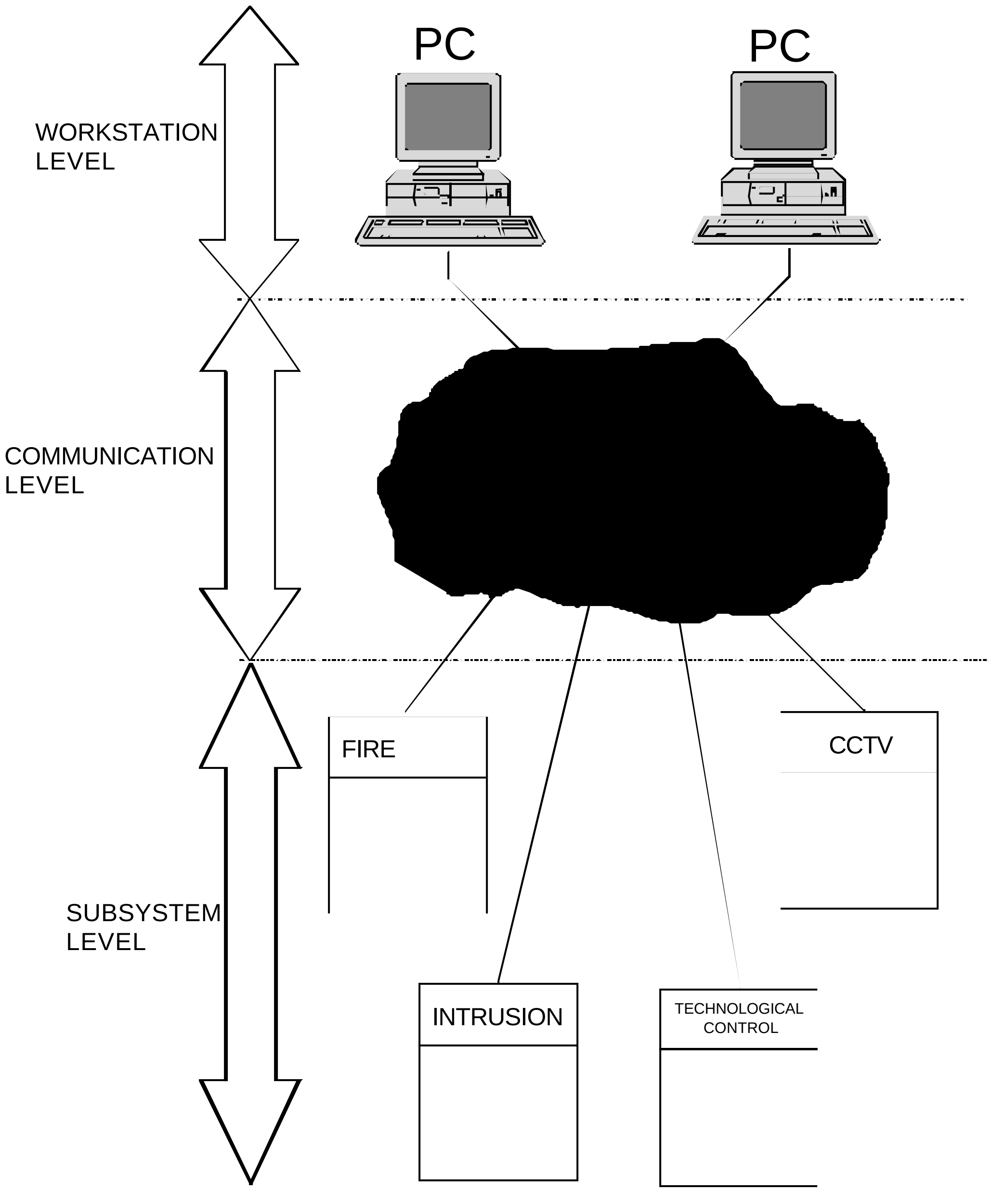
1 INTRODUCTION

LMS alarm is an Alarm Monitoring System (LMS stands for Local Monitoring System) designed for an effective and easy central management of a large set of safety and security devices.

An LMS modular systems is made up of:

- PC based workstation(s) equipped with Windows application software
- Communication network devices
- Safety and security subsystems (e.g. Fire detection, Intrusion detection, CCTV, Technological control)

LMS modular provides the operators a common, easy to operate interface that unifies the handling of all the safety and security related alarm events.



The PC software, based on Microsoft Windows, is a Cerberus developed applications, especially studied for user friendly, menu-driven operations. The flexible structure of the LMS modular software and hardware architecture allows the integration of new types of subsystems without major efforts.

Scope of this document is to provide clear technical guidelines to engineers interested to integrate a new subsystem-level device in an LMS modular system. Please note that relay-contact interfacing will not be discussed here, even though this approach is very well possible whenever the cost of a communication software development cannot be budgeted.

Instead, we will focus on the ay to implement a serial-line communication that allows a powerful and fast data exchange between LMS modular and the subsystem.

The physical link and the data-link protocol issues are briefly presented in section 2 and 3 in order to give a complete introduction. Please note that these subjects are better described in the related documents [1] and [2].

The following sections deal with the application level protocol which is deeply discussed. Section 4 illustrates the image database for foreign subsystem modelling and section 5 presents the LMS modular control functions. The application frame messages are then described in Section 6.

1.1 Related Documentation

[1]	LMS modular Engineering Guidelines	e1139
[2]	CDDL- Cerberus Dati Data linkprotocol	e1152
[3]	CDSF- Cerberus Dati Standard Format application protocol	e1151
[4]	GW-00 Technical Manual	e1114
[5]	GW-01 Technical Manual	e1112
[6]	LMS modular User Manual	e1109

2 PHYSICAL LEVEL

2.1 Serial line characteristics

The physical link between LMS modular and the subsystem is a RS-232c asynchronous serial interface line with data leads only: TX, RX and GROUND.

Serial line must be dedicated and permanently available, no switched line can be used.

No control of modem signals (RTS, CTS, DTR, DSR, DCD) is provided. However, modems can be used and, being the line permanently active, control signals can be jumpered whenever a modem needs them. See connection examples in 4 and 5.

Asynchronous parameter setting is: 8 bits, no parity, 1 stop bit. Supported speeds are: 300, 600, 1200, 2400, 4800, 9600 bauds.

2.2 Architectures

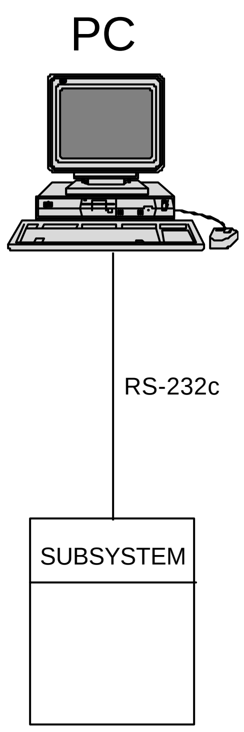


Fig. 2.1

The subsystem can be directly connected to the workstation PC (fig. 2.1) for a one-to-one architecture.

Also, a communication gateway can be placed in-between so as to allow the centralization of more subsystems onto one or more workstation (fig. 2.2) Gateways can have two configurations: GW-00 for up to 20 subsystems and 4

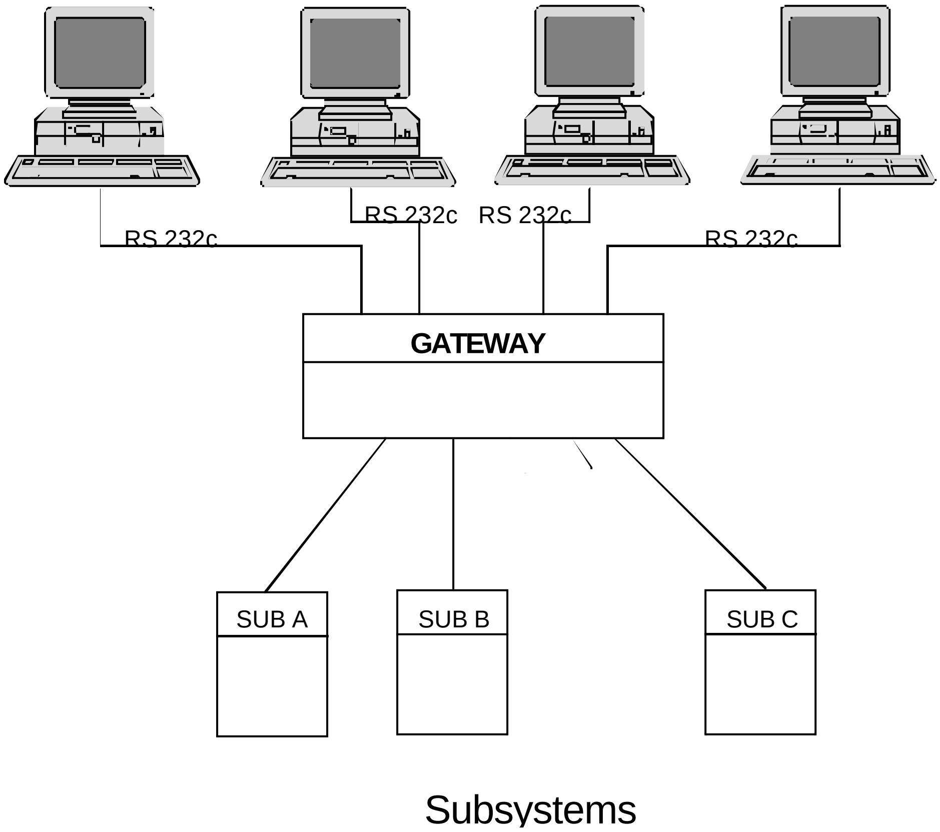


Fig. 2.2

workstations (see [4]) and GW-01 for up to 4 subsystems and 2 workstations (see [5]).

More sophisticated architectures allow a higher connectivity to LMS modular system. A network of gateways (see fig. 2.3) can be setup when a large number of subsystems has to be connected.

Please refer to [1] for a detailed description of the possible architectures.

3 DATA LINK LEVEL

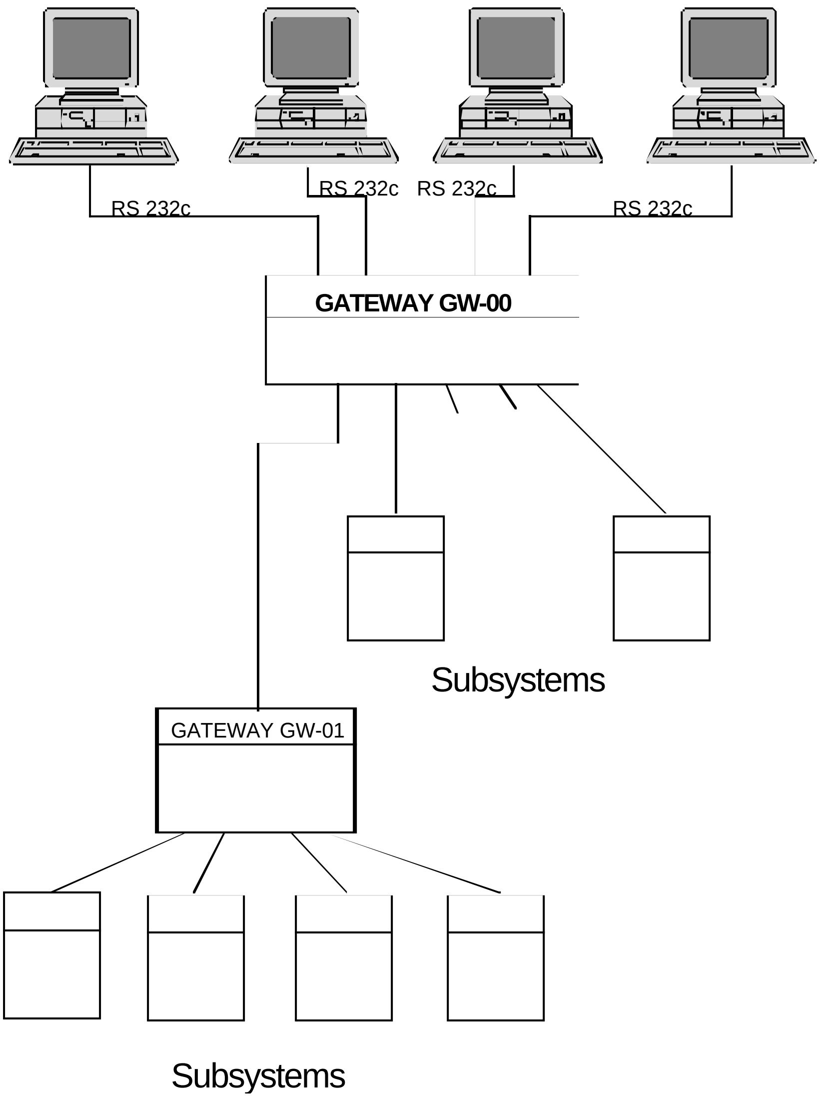


Fig. 2.3

In any of the architectures depicted in the previous section, data is transferred to and from the subsystem with the CDDL data link protocol, which is extensively described in [2].

In general, CDDL can be indicated as a point-to-point, byte oriented, polling-selecting, binary protocol.

4 APPLICATION LEVEL

4.1 Subsystem information model

The information transmitted by the subsystem to LMS modular must allow the monitoring workstation to generate a complete representation model of the subsystem process.

LMS modular is not only a logger of what goes on in the subsystem connected to it. Instead, it handles a real-time image database which represents in any moment the physical events in progress.

Also, LMS modular can issue control commands in order to modify the subsystem states or behaviour. Figure 1 shows the information flow.

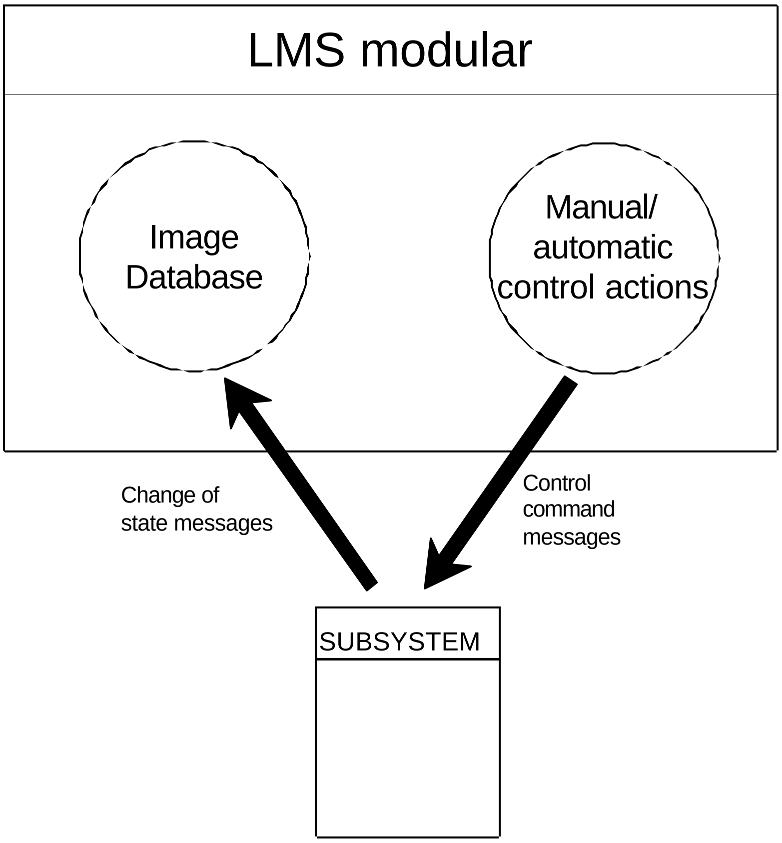


Fig. 4.1

The "change of state" message, transmitted by the subsystem, cause a modification in the LMS modular Image database. The control commands are instead generated by the monitoring workstations, with manual or automatic operations.

4.2 Point database

LMS modular database collects the subsystem image information in multi-state datapoints. Each subsystem is modelled in LMS modular by a specific set of multi-state point. Each "change of state" message coming from the subsystem is interpreted by LMS modular and results in a change of state of the datapoint devoted to the representation of that specific piece of information. (refer to Fig. 4.2).

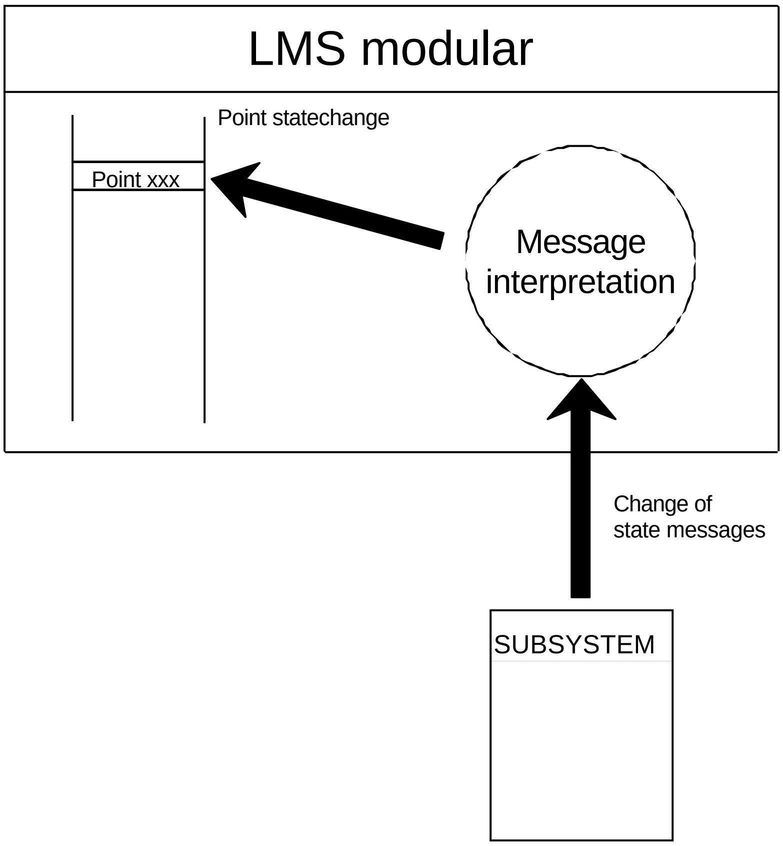


Fig. 4.2

4.3 Point states

Each datapoint should fully represent a precise physical or logical part of the subsystem process, for example a security detector, an intrusion protected zone or a power supply unit. LMS modular point can assume 16 different state values, numbered from 0 to 15. It is important to stress that the states must be mutually exclusive, that is, only one at the time must be set at any given moment: a detector can be active or not, but it cannot assume both states at the same time.

Each point state should represent a specific condition of the subsystem process in a given moment of time., Fig. 4.3 illustrates a simple example of point state diagram.

The "power supply" datapoint can assume three different states correspondent to three different conditions in the subsystem's power supply unit.

In the diagram, the "bubbles" indicate the point states and the "arrows" represent as change of state message issued by the subsystem.

We will see in the next sections how the subsystem can transfer the change of state information to LMS modular.

However, we can now state that the subsystem is expected to generate the specific message whenever a change of its condition occurs.

In this message it must be indicated:

1) Which is the part of the subsystem involved (= **which is the data point**)
2) Which is the condition reached by this part (= **which is the point state value**)

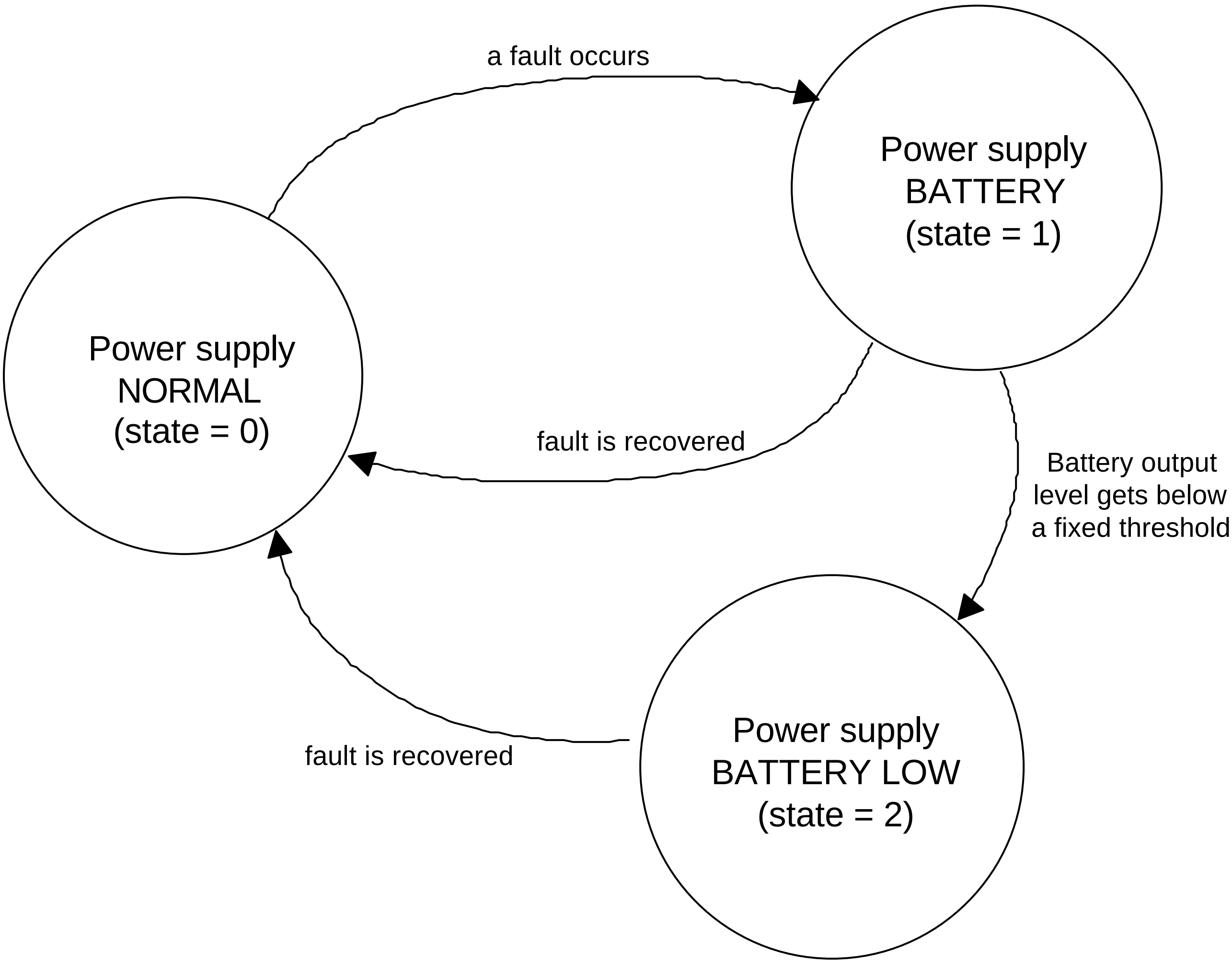


Fig. 4.3

See section 6 for the exact content of change of state messages.

4.4 Subsystem model design

It is now clear how to design a **multi-state point model** of a subsystem:

1) we need to list all the subsystem parts, components, elements that deserve to be presented by LMS modular workstations to the final users.
   This list results to be the actual point list where the subsystem process image will be stored. If necessary, up to 8 different lists of point can be supported so as to match different versions of models of the subsystem.
2) for each point, all the possible states should be documented. The maximum number of states supported is 16 although this limit should never be reached. A point whose state can assume 16 values will probably be too complicated. Whenever a high number of states becomes necessary (more than 10, as a thumb rule), the generation of a new point (so as to split up states into 2 points) must be evaluated.
   Some examples of subsystem modelling are given in appendix A.
3) points can be collected in homogeneous groups so as to be displayed in meaningful views in LMS modular workstations.
   The point list should therefore include a "view" option that indicates which points have to be grouped together.
   Some examples of subsystem modelling are given in appendix A.

5 LMS MODULAR SUBSYSTEM HANDLING

5.1 LMS modular presentation of database information

As described above, LMS modular is able to handle a point database that can store a process image based on multi-state points.

Also, the LMS modular workstation provides an effective graphic and alphanumeric display of the datapoints so as to dynamically present to the final users the evolution of the field-level process on a user-friendly interface (see Fig. 5.1).

Point state values are translated by LMS modular software into clear texts or graphic symbols in a way easily understandable by operators.

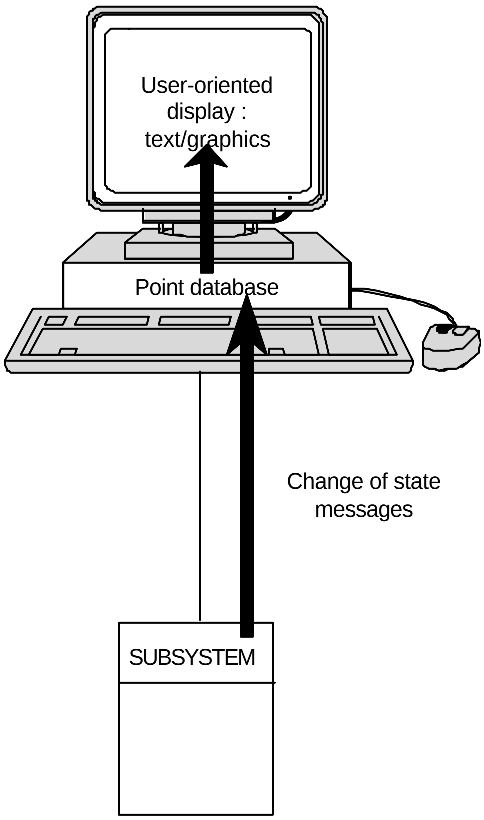


Fig. 5.1

5.2 LMS modular process control

Besides the pure representation of the information on the screen, LMS modular can provide control functions that allow operators to modify the conditions of the subsystem process.

These control functions can be commands meant to:

1) modify the status or the behaviour of the subsystem: general control commands;
2) react to a specific event message so as to handle danger or a somehow abnormal conditions: event acknowledgement and reset commands

5.2.1 Control commands

LMS modular workstation provides the capability of generating subsystem's control commands upon both manual action and automatic time-driven programmed control.

As it is shown in Fig. 4.1 control messages are transmitted to the subsystem.

These control messages contain:

1) the type of action to be performed (= **command number**)
2) the specific part of the subsystem involved in the action (= **point number parameter**)
3) additional parameter(s) concerning the specific command action

Appendix B contains some examples of control command lists. Actually, part of the subsystem model design is to define which control commands**Fehler! Textmarke nicht definiert.** have to be available. Section 6 shows the control command frames.

5.2.2 Event acknowledgement and reset

LMS modular can be configured in order to handle certain change of state messages in a special way so as to get the operator's attention on what occurred.

First, the workstation can automatically issue an acknowledgement command whenever the operator take into account the event displayed onto the screen.

This acknowledgement (ack) command can be used by the subsystem to modify its behaviour because it can be assumed that someone is now in charge of treating the abnormal condition indicated by the initial change of state message.

For example, if the subsystem has somehow detected a security alarm and consequently activated a local buzzer, it can silence the buzzer upon receiving the ack command.

Additionally, LMS modular workstation is capable to reset a certain event a specific command so as to clear up a latched information at subsystem level.

For instance, if a fire alarm was detected and latched by subsystem, it can be cleared with a reset command.

The subsystem may or may not execute the reset command (as well as the ack one) depending on the current condition: for istance, if an intrusion detector is still active, the reset will not clear up the alarm in the subsystem.

The ack function allows the subsystem to be informed about a higher-level treatment of an event, the reset functions permits to delete past latched event in the subsystem memory. Ack and reset can be used individually or combined together for a complete event treatment.

An example of an alarm treatment is given in figures 5.2 and 5.3.

In Fig. 5.2 the sequence of messages is depicted: the subsystem informs LMS modular about the alarm events and also about the execution of ack and reset commands. Fig. 5.3 illustrates the point state diagram.

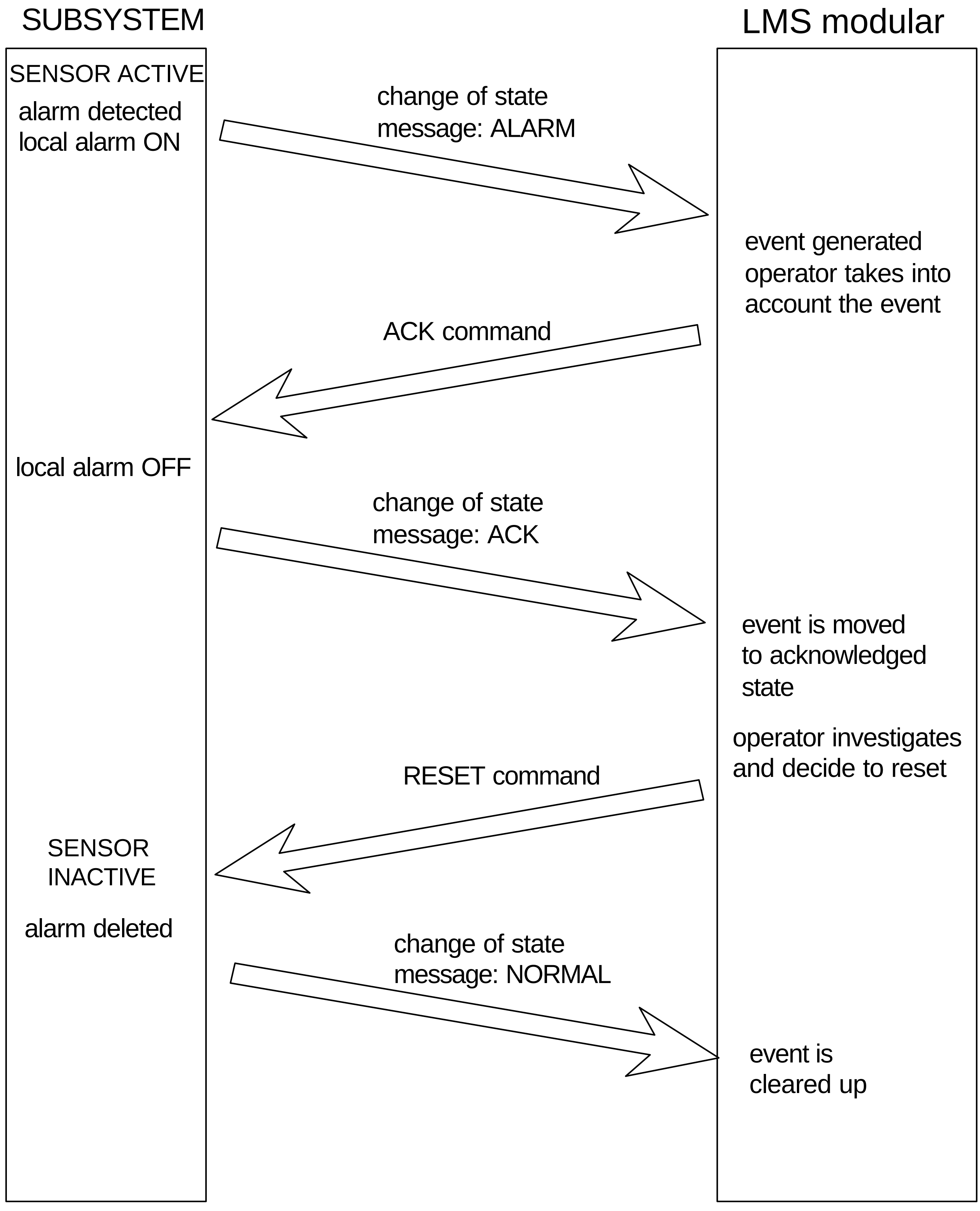


Fig. 5.2

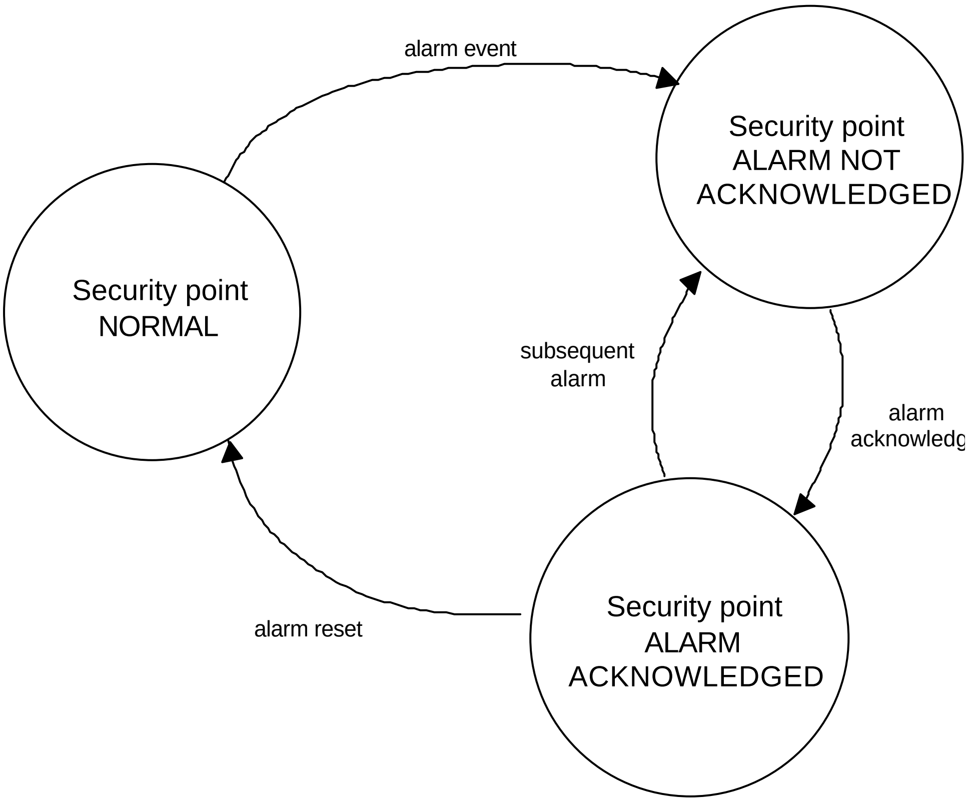


Fig. 5.3

In order to allow a complete control of the ack and reset functions, LMS modular handles a specific attribute information associated to each point state.

Basically, besides the 0 to 15 point state value, the change of state" message (and consequently the LMS modular database) also contains a **treatment flag** that indicates whether that specific state is:

- normal, i.e. any pending event can be deleted
- abnormal, and ack is necessary or not
- abnormal, and reset is necessary or not

The table below shows the complete list of cases for the LMS modular treatment flag.

Value	Status	Behaviour
0	normal	no acknowledgement command is issued; no reset command is issued; LMS modular can delete any pending event.
1	abnormal	not to be acknowledged, not to be reset; no acknowledgement command is issued; no reset command is issued.
2	abnormal	to be acknowledged, not to be reset; acknowledgementcommand is issued; no reset command is issued;
3	abnormal	not to be acknowledged, to be reset; no acknowledgement command is issued; reset command is issued;
4	abnormal	to be acknowledged, to be reset; acknowledgement command is issued; reset command is issued

In figs. 5.4 and 5.5, the examples of fig. 4.3 and 5.3 are presented again, with the new Treatment Flag information added.

POWER SUPPLY POINT
(no ACK / no RESET needed)

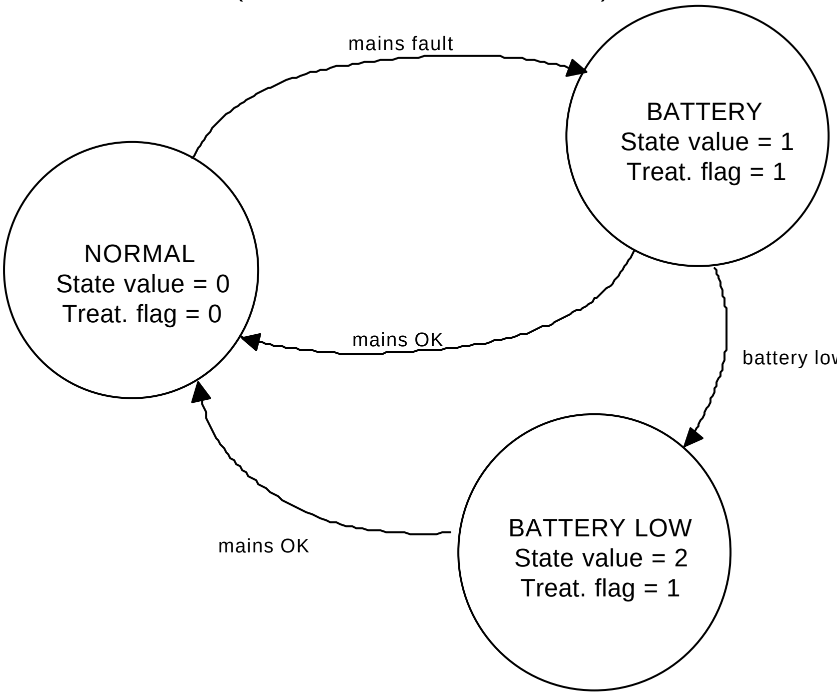


Fig. 5.4

SECURITY ALARM POINT
(ACK and RESET needed)

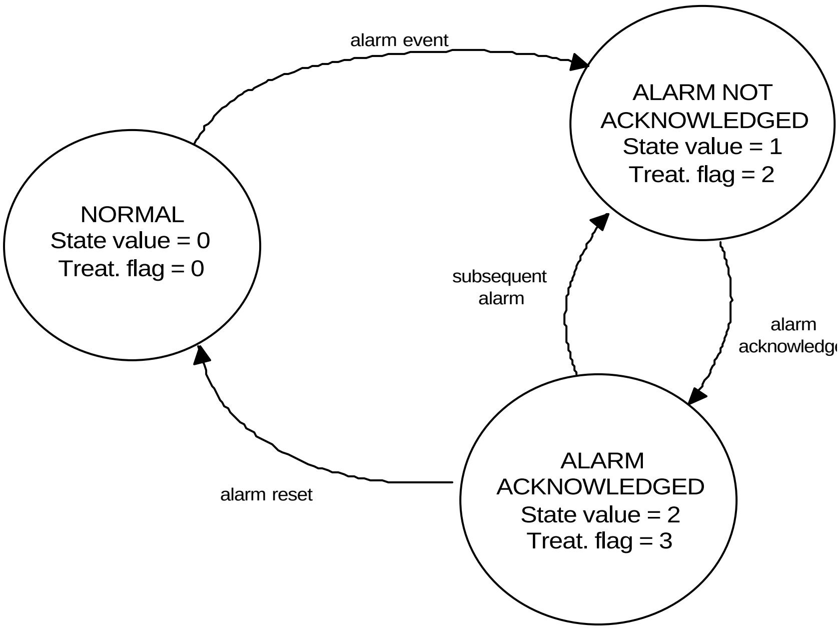


Fig. 5.5

6 CDSF PROTOCOL

The Cerberus Dati Standard Format (CDSF) protocol, described in [3]; supports the application message functions above described in section 4 and 5.

Both "Change of state" and "control command" messages are available as 16 byte long, CDSF frames.

6.1 "Change of state" frame

Here is the structure of the 16 byte CDSF message frame

byte	0	0
	1	Year (two BCD figures)
	2	Month (two BCD figures)
	3	Day (two BCD figures)
	4	Hour (two BCD figures)
	5	Minutes (two BCD figures)
	6	Seconds (two BCD figures)
	7	Subsystem cluster address LSB (binary value)
	8	Subsystem cluster address MSB (binary value)
	9	Data Point number LSB (binary value)
	10	Data Point number MSB (binary value)
	11	Point state treatment flag (binary value):
	12-14	0
	15	Point state value (binary value):

Notes:

1. byte 0 = 0 means digital message
2. bytes 1-6 contain the time stamp related to the change of state
3. bytes 7-8 indicate a subsystem address. This is usually 0, but it can be used to distinguish a subsystem in a (virtual or real) cluster of devices
4. bytes 9-10 contain the datapoint number described in section 4.3
5. byte 11 is the treatment flag mentioned in section 5.2.2
6. byte 15 contains the datapoint state value (0 to 15) as indicated in section 4.3

here is an example of message frame. It refers to point 13 that changed to value 7 on Jan 21st, 1994 at 13:04:55 (all values are in hexadecimal)

Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Content	00	94	01	21	13	04	55	00	00	0D	00	01	00	00	00	07

6.2 "Control command" frame

Here is the structure of the 16 byte CDSF command frame

byte	0	80 hex
	1	Subsystem cluster address LSB (binary value)
	2	Subsystem cluster address MSB (binary value)
	3	Command number LSB (binary value)
	4	Command number MSB (binary value)
	5	Command first parameter LSB (binary value)
	6	Command first parameter MSB (binary value)
	7	Command second parameter LSB (binary value)
	8	Command second parameter MSB (binary value)
	9	Command third parameter LSB (binary value)
	10	Command third parameter MSB (binary value)
	11	Command fourth parameter LSB (binary value)
	12	Command fourth parameter MSB (binary value)
	13	Command fifth parameter LSB (binary value)
	14	Command fifth parameter MSB (binary value)
	15	0

Notes:

1. byte 0 = 80 hex means digital command
2. bytes 1-2 indicate a subsystem address. This is usually 0, but it can be used to distinguish a subsystem in a (virtual or real) cluster of devices
3. bytes 3-4 contain the command number indicating the specific control action to be executed
4. bytes 5-14 contain 5 16 bit parameters that may contain additional information; parameters not used must be set to 0
6. byte 15 is not used

7 ANNEX A - EXAMPLES OF SUBSYSTEM MODELLING

7.1 A1. Fire detection unit

This example deals with a 3 zone fire detection control panel. The subsystem is modeled as described in the following table: Please note that the meaning of points 1 and 2 is the same for any subsystem.

Point no	Description	Status	1	2	3	4
		Value	0	1	2	3
1	Communication link		NORMAL	FAULT	-	-
2	Out_of_scan status		NORMAL	OUT OF SCAN		
3	General_fault		NORMAL	FAULT	-	-
4	Power_supply		NORMAL	MAINS OFF	BATTERY LOW	-
5	Fire_alarm_zone_1		NORMAL	ALARM	FAULT	EXCLUDED
6	Fire_alarm_zone_2		NORMAL	ALARM	FAULT	EXCLUDED
7	Fire_alarm_zone_3		NORMAL	ALARM	FAULT	EXCLUDED

7.2 A2. CCTV unit

This example deals with a CCTV (Closed Circuit Television) control panel. The subsystem is modeled as described in the following table. Please note that the meaning of points 1 and 2 is the same for any subsystem.

Point no	Description	Status	1	2	3	4
		Value	0	1	2	3
1	Communication link		NORMAL	FAULT	-	-
2	Out_of_scan status		NORMAL	OUT OF SCAN		
3	General_fault detection		NORMAL	FAULT	-	-
4	Power_supply		NORMAL	MAINS OFF	BATTERY LOW	-
5	Satellite unit status		NORMAL	FAULT	LINE FAULT	
6	VCR status		NORMAL	FAULT	TAPE END	
7	Motion detection		NORMAL	ALARM		

8 ANNEX B - EXAMPLES OF CONTROL COMMAND LISTS

8.1 B1. Fire detection unit

Command no	Description	1	2	3	4	5	Point(s)
0	Status request	0	0	0	0	0	all
0	Date/time synchronization	1	YYMM	DDHH	MMSS	0	-
1	Acknowledgment	point no or 0=general	0	0	0	0	any
2	Reset	point no or 0=general	0	0	0	0	any
3	Exclude zone	point no.	0	0	0	0	zones
4	Include zone	point no.	0	0	0	0	zones

8.2 B2. CCTV unit

Command no	Description	1	2	3	4	5	Point(s)
0	Status request	0	0	0	0	0	all
0	Date/time synchronization	1	YYMM	DDHH	MMSS	0	-
1	Acknowledgment	point no or 0=general	0	0	0	0	any
2	Reset	point no or 0=general	0	0	0	0	any
3	Exclude zone	point no.	0	0	0	0	zones
4	Include zone	point no.	0	0	0	0	zones

9 ANNEX C - SUBSYSTEM MODEL DESIGN CHECKLIST

#	Descriptions	Database limits
1	Multi-state datapoint list	32000 points 8 lists for type
2	Datapoint states	16 states for point
3	Command list	
4	Command parameters	5 parameters for command (1st = point number)
5	Datapoint views in management	